String kernels for complex time-series: Counting targets from sensed movement

Theodoros Damoulas*, Jin He†, Rich Bernstein‡, Carla P. Gomes‡ and Anish Arora†
*Center for Urban Science + Progress (CUSP), New York University, Brooklyn, NY, USA, damoulas@nyu.edu
†Computer Science and Engineering, Ohio State University, Columbus, Ohio, USA, he.146@osu.edu, anish@cse.ohio-state.edu
‡Department of Computer Science, Cornell University, Ithaca, NY, USA, rab38@cornell.edu, gomes@cs.cornell.edu

***Abstract*—Complex (imaginary) signals arise commonly in the field of communications in the form of time series in the complex space. In this work we propose a symbolic approach for such signals based on *string* kernels derived from a complex SAX representation and apply it to a challenging counting problem. Our approach, that we call `cStrings`, is within a Gaussian process regression framework and outperforms established Fourier transforms and complex kernels, achieving a correlation coefficient of 0.985 when predicting the number of targets sensed by a pulsed Doppler radar.**

I. Introduction

In telecommunications, electromagnetics, and radar technology it is very natural to encounter sensory signals that are complex time series: $\mathbf{x}_{1:T} \triangleq \{x_t\}_{t=1}^{T} \in \mathbb{C}^T$ where $x_t = \{\alpha_t + \beta_t i \in \mathbb{C} \mid \alpha_t, \beta_t \in \mathbb{R}\}$. For example consider the sensory output of the low-power mote-scale BumbleBee sensor, Figure 1, that can be employed to count the number of targets in a spatio-temporal region S. This sensor has applications in various fields such as environmental monitoring, sustainability, security and defense, and in its highest sensitivity it can detect human heartbeats within its range.

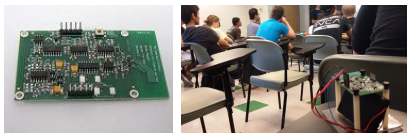

Fig. 1. The BumbleBee sensor is a monostatic wideband radar, with a bandwidth of 100MHz, a center frequency of 5.8GHz and a maximum detection range for people at ∼10m omnidirectional.

The sensory complex output x_t contains information regarding movement, direction of movement and distance to the sensor. Since that information is summarized by the sensory output every time step t the task of modeling and predicting the number of moving targets, or the total number of targets y is very challenging. Assuming some stationarity of the process, namely that y is constant within S, gives rise to a time series data $\mathbf{x}_{1:T} \in \mathbb{C}^T$ for predicting the number of targets y in S.

The motivation for our work comes from the sustainability domain where this sensor will be employed to count human and animal targets within the Panna national park in India. This is part of a larger integrated wireless sensor network that will be in place for the protection of the tiger population and monitoring of illegal human activities like poaching. In this work we attempt to predict the number of students in a classroom given a 30 sec BumbleBee measurement of activity, Figure 2(a).

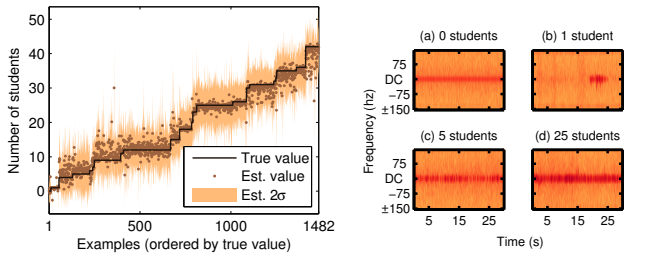

Fig. 2. (a) Predicting the number of human targets (students) in a classroom with the proposed approach based on the complex time series data from the BumbleBee sensor. (b) Typical (log) power spectrum for the radar.

The problem can be stated as: *Given a complex vector that is a time-series $\mathbf{x}_{1:T} \in \mathbb{C}^T$ describing the number of moving targets at any time t, predict the total number of targets y within the time frame T.* In this work we are going to focus on direct prediction of y but we note that in terms of inference this problem has a latent variable representation.

II. Related work

A. Target (people) counting

People counting has been a significant line of research within computer vision [1], [2], [3] but has not been adressed until now with radar based approaches. Considering the deployment domains of interest and their restrictions such as resource constraints, limited power, occlusion and low-light environments it is straightforward to see that computer vision based systems are ill-suited for this task. Furthermore, the natural underperformance of computer vision systems under occlusion and night settings highlights the need for alternative sensing systems and corresponding machine learning techniques.

B. Complex signals and time series

Techniques for analyzing (complex) time series have been traditionally studied within the field of discrete signal processing [4] and computer science [5], [6], [7]. The main tools are the discrete Fourier and wavelet transforms (DFT, DWT) that both offer orthogonal basis in the Hilbert space $\mathcal{H}$ of the function f analyzed. These are complimented by other approaches such as dynamic time warping (DTW) [8], [9], hidden Markov models (HMMs) [10], Kalman filters [11],

symbolic representations such as SAX [12], [13], [14] and piecewise linear techniques [15], [16]. In terms of analyzing complex time series $\mathbf{x}_{1:T} \in \mathbb{C}^T$ some of the representations such as DFT extend naturally to $\mathbb{C}^T$, and we will make use of that, while others do not or require intermmediate DFT estimation. Before proceeding to the proposed symbolic approach we briefly review some relevant work on complex valued signals via complex Gaussian kernels and corresponding supervised learning algorithms.

Complex kernels Recently, [17] proposed algorithms on *complexification* of real *reproducing kernel hilbert spaces* (RKHS) and complex kernels for Least Mean Squares (LMS) and Support Vector Regression (SVR). The following Lemma holds:

Lemma II.1. [17]
If $k_{\mathbb{C}}(\mathbf{x}, \mathbf{x}')$ is a complex kernel, then

$$k_{\mathbb{R}}\left(\begin{pmatrix}\mathbf{x}^r\\ \mathbf{x}^i\end{pmatrix}, \begin{pmatrix}\mathbf{x}'^r\\ \mathbf{x}'^i\end{pmatrix}\right) = \mathrm{Re}(k_{\mathbb{C}}(\mathbf{x}, \mathbf{x}')) \tag{1}$$

where $\mathbf{x}^r$, $\mathbf{x}'^r$, $\mathbf{x}^i$, $\mathbf{x}'^i$ are the real and imaginary parts of $\mathbf{x}$, $\mathbf{x}' \in \mathbb{C}^d$, is a real kernel.

Proof: The proof is given in [18] ■

The above states that a complex kernel $k_{\mathbb{C}}$ can be shown to induce a real RKHS $\mathcal{H}$ with kernel $k_{\mathbb{R}} \in \mathbb{R}^{2d} \times \mathbb{R}^{2d}$. A Gaussian complex kernel for the complex time-series data $\mathbf{x}_{1:T} \in \mathbb{C}^T$ is then given by:

$$k_{\mathbb{C}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}'^{\star})^{\mathrm{T}} \mathbf{\Lambda} (\mathbf{x} - \mathbf{x}'^{\star})\right) \tag{2}$$

where * denotes the complex conjugate and $\mathbf{\Lambda}$ is the diagonal matrix containing the length-scale hyperparameters $\boldsymbol{\theta}$. The inclusion of per-dimension scale hyperparameters is called automatic relevance determination (ARD). The induced real kernel $k_{\mathbb{R}}$ describes both in-phase and quadrature components of $\mathbf{x}_{1:T} \in \mathbb{C}^T$ and is *not* equivalent to the standard real Gaussian kernel, as can be seen from the case $\mathbf{x} = \mathbf{x}'$.

The complex time series from the sensor i) have no possible *time alignment*, i.e. the target movements across different samples are spontaneous and chaotic, ii) might arise from uncorrelated, *independent movements*, i.e. is not clear if 1st order Markov assumptions can be useful and to what extend, iii) have no *distance to source* information for appropriate scaling, i.e. the power intensity of the signal is inversely proportional to the fourth power of distance but such information is not readily available for signal normalization, and finally iv) the *signal-to-noise* ratio varies as a function of movement speed due to low frequency radar noise, hence low speed activities cannot easily be detected.

Considering the above problem characteristics and the underlying modeling assumptions from the reviewed approaches it is clear that only a subset of techniques can be employed for experimental comparison against the following approach that we propose in this paper.

C. Time series approximations

We will extend two discrete approximation methods for time series to the complex-valued domain: the Piecewise Aggregate Approximation (PAA) [16], [19] and the Symbolic Aggregate approXimation (SAX) [12]. PAA is a dimension reduction technique where each non-overlapping window of a fixed size in the sequence is replaced by the arithmetic mean of the data within that window. The SAX approximation converts the series of continuous-valued elements to discrete-valued symbols based on equiprobable regions. The equations for these transformations presented as Eq. 3 and Eq. 5 in the following section reduce to the original form when applied to real-valued sequences.

III. String kernels for complex time series

The idea behind `cStrings` is to transform the complex time series into a symbolic representation by extending the Piecewise Aggregate Approximation (PAA) and the Symbolic Aggregate approXimation (SAX) to handle complex vectors, and then construct string kernels based on n-gram representations of the signals, see Figure 3 and Figure 4. This allows us to capture informative subsequences, n-grams, in the complex time series that are position independent and predictive of the total number of targets within S.

A. Symbolic representation of the complex domain

Consider a complex time series $\mathbf{x}_{1:T} \triangleq \{x_t\}_{t=1}^T \in \mathbb{C}^T$ where $x_t = \{\alpha_t + \beta_t i \in \mathbb{C} \mid \alpha_t, \beta_t \in \mathbb{R}\}$ and a set of N such series with associated counts $\{\mathbf{x}_{n,1:T}, y_n\}_{n=1}^N$. We can extend PAA and SAX to the complex domain and represent a complex time series $\mathbf{x}_{1:T}$ as a D-dimensional ($D \ll T$) vector with elements:

$$\tilde{x}_i = \frac{D}{T} \sum_{j=\frac{T}{D}(i-1)+1}^{\frac{T}{D}i} x_j \tag{3}$$

Lower bound on distance The new representation $\tilde{\mathbf{x}}_{n,1:D} \in \mathbb{C}^D$ retains a lower bound on $\mathcal{L}_p$ measures $\mathcal{L}_p(\tilde{\mathbf{x}}_{n,1:D}, \tilde{\mathbf{x}'}_{n,1:D}) \leq \mathcal{L}_p(\mathbf{x}_{1:T}, \mathbf{x}'_{1:T})$ with respect to the original representation $\mathbf{x}_{n,1:T} \in \mathbb{C}^T$ where:

$$\mathcal{L}_p(\mathbf{x}_{1:T}, \mathbf{x}'_{1:T}) = \left(\sum_{t=1}^{T} |x_t - x'_t|^p\right)^{\frac{1}{p}} \tag{4}$$

Theorem III.1. *For any sequence $\mathbf{x}_{1:T} = [x_1, \ldots, x_T] \in \mathbb{C}^T$ and $1 \leq p < \infty$ the following holds:*

$$\mathcal{L}_p(\tilde{\mathbf{x}}_{n,1:D}) \leq \mathcal{L}_p(\mathbf{x}_{1:T})$$

Proof: We know that $f = |\cdot|^p$ for $1 \leq p < \infty$ is a convex function in $\mathbb{R}^{2\nu}$ and hence it is also convex in $\mathbb{C}^\nu$. Following [16] if we define $\lambda_1, \ldots, \lambda_L \in \mathbb{R}$ such that $\lambda_i \geq 0$ and $\sum_{i=1}^L \lambda_i = 1$ then it follows that

$$f(\lambda_1 x_1 + \ldots \lambda_L x_L) \leq \lambda_1 f(x_1) + \cdots + \lambda_L f(x_L)$$

and substituting for $\lambda_i = \frac{1}{L}$, $f = |\cdot|^p$ and rearranging based on Eq. 3 completes the proof. ■

We can now discretize the PAA representation further by defining $(\mathcal{N}(0,1))$ equiprobable breakpoints along both real and imaginary axis and extend SAX [12] to the complex domain with a 2D alphabet, Figure 3. This implies that the new SAX mapping is:

$$\begin{aligned} \hat{c}_i = \mathbf{A}_{j,k} \iff & \quad \beta_{j-1} \leq \mathrm{Re}(\tilde{x}_i) < \beta_j \\ \wedge & \quad \beta_{k-1} \leq \mathrm{Im}(\tilde{x}_i) < \beta_k \end{aligned} \tag{5}$$

where $\mathbf{A}_{j,k}$ is the 2D symbolic alphabet and β are the standard equiprobable breakpoints from the Gaussian CDF.

B. *n-grams string kernel*

Having obtained a symbolic representation of the complex domain where the time series are defined we can now adopt a text mining perspective and construct appropriate string kernels [20], [21]. We consider n-grams that map the time series into high-dimensional feature vectors $\nu() \in \mathbb{R}^V$, where each dimension describes the number of occurrences of a given contiguous subsequence. The number of distinict n-grams V from alphabet $\mathbf{A}$ is bounded by $V \leq |\mathbf{A}|^n$. In practise the representation is very sparse as not all n-grams are represented in every time series. Finally the n-gram feature vector is normalized using the term frequency-inverse document frequency statistic (*tfidf*) commonly used in text mining applications to discount the contribution from common subsequences:

$$tfidf\,(v, \mathbf{x}_i, \mathbf{X}) = c(v, \mathbf{x}_i) \log \frac{N}{|\mathbf{x}_j \in \mathbf{X} : v \in \mathbf{x}_j|} \tag{6}$$

where $c(v, \mathbf{x}_i)$ is the raw count of n-gram v in the sequence, and N is the number of sequences. The string kernel (p.s.d. by construction) between two complex time series $\mathbf{x}_{1:T}, \mathbf{x}'_{1:T} \in \mathbb{C}^T$ is then given by:

$$k(\mathbf{x}_{1:T}, \mathbf{x}'_{1:T}) = \frac{\nu(\hat{\mathbf{c}})^\intercal \nu(\hat{\mathbf{c}}')}{\|\nu(\hat{\mathbf{c}})\|_2 \|\nu(\hat{\mathbf{c}}')\|_2} + b \tag{7}$$

where $b \in \mathbb{R}$ is a bias constant, $\nu()$ is the vector of n-gram counts following the tranformations defined in Eq. 3, 5, and 6, and the denomitator denotes scaling to obtain a cosine kernel. The sparse n-gram string kernel has a linear time complexity [21] and hence is a viable solution for real-time deployment.

C. *Gaussian process regression*

In order to quantify the uncertainty of our predictions we turn to Bayesian inference and employ Gaussian process regression (GPR) as the kernel based learning algorithm. The GP prior is specified by the mean $\mathbf{m}$ and covariance kernel function $k(x, x')$:

$$f(x) \sim \mathcal{GP}\left(\mathbf{m}(x), k(x, x')\right) \tag{8}$$

We assume a prior with zero mean and posterior inference in GPR is available in closed form due to the Gaussian nature of the prior and likelihood, see [22]. Hyperparameter optimization for the complex and real Gaussian kernel is done via conjugate gradients and the partial derivatives of the marginal likelihood $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$ w.r.t. the hyperparameters $\boldsymbol{\theta}$ are given by:

$$\begin{aligned} &\frac{\partial}{\partial \theta_j} \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \\ &\frac{1}{2}\mathbf{y}^\intercal K^{-1} \frac{\partial}{\partial \theta_j} K^{-1}\mathbf{y} - \frac{1}{2}\mathrm{trace}\left(K^{-1}\frac{\partial K}{\partial \theta_j}\right) \end{aligned} \tag{9}$$

Where the derivatives $\frac{\partial K}{\partial \theta_j}$ for the complex kernel K follow exactly as the real double exponential case. The overall proposed algorithm is given in Algorithm 1.

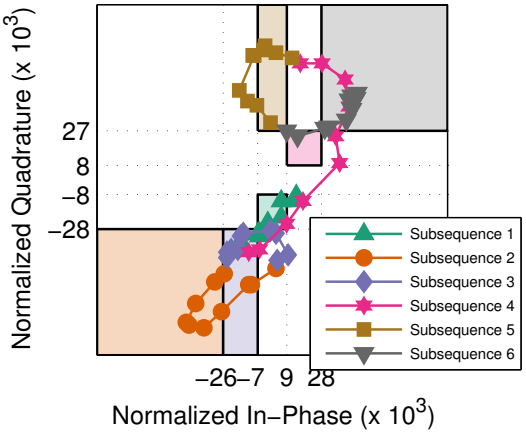

Fig. 3. 2D aggregation, SAX and n-gram construction. Every marker denotes a complex number of the time series, colored cells denote the SAX representation of the corresponding 10 complex numbers that gives rise to the alphabet letter. The alphabet size (number of equiprobable regions) in this example is 5x5 and this sequence gives rise to a 6-gram. See also Figure 4 for the correpsonding representation in time.

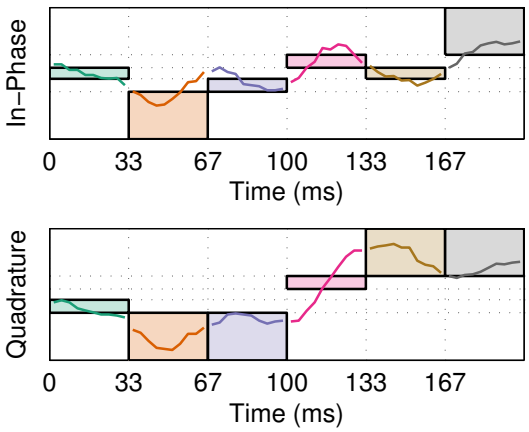

Fig. 4. Time-series representation of the in-phase and quadrature components from Figure 3

Algorithm 1 `cStrings`: Complex String kernels

Input: Time series and targets $\{\mathbf{x}_{n,1:T}, y_n\}_{n=1}^N$
Initialize by CV: Window T/D, $|\mathbf{A}|$, n-gram length n
for $n = 1$ **to** N **do**
  for $i = 1$ **to** D **do**
    (Complex PAA): $\tilde{x}_{n,i} \leftarrow$ Eq. 3
    (Complex SAX): $\tilde{c}_{n,i} \leftarrow$ Eq. 5
  end for
  Index n-grams $\nu(\hat{\mathbf{c}})$
  Normalize ν by *tfidf*
  for $j = 1$ **to** n **do**
    $k(\mathbf{x}_{n,1:T}, \mathbf{x}'_{j,1:T}) \leftarrow$ Eq. 7
  end for
end for
$\mathcal{GP} \leftarrow K, \mathbf{y}$ (Exact inference)

D. Computational complexity

The time complexity is a function of the number of samples N, length of orginal time series T and length of symbolic representation D. The GPR, as most kernel machines, has a dominant term $\mathcal{O}(N^3)$ that can be improved by sparsity inducing priors and Nyström approximations [23]. The proposed string kernel has a dominant term $\mathcal{O}(DN^2)$ while the total PAA complexity is linear $\mathcal{O}(NTD)$. The overall string kernel construction can be brought down to a linear time complexity [21] while analogous computational improvements are available for SAX [8] and can be exploited within this framework. Finally it is worth noting that storage requirements and space compexity are significantly reduced via the symbolic representation [13] in relation to DFT approaches.

IV. Datasets, experimental setting and methods

In this section we describe the real world experimental setting, resulting data and competing methodologies that we compare against the proposed approach. All the experiments were conducted with the BumbleBee sensor placed in four different classrooms with a varying number of students (0-40) and while performing various activities (teaching, breaks, tutorials, etc.), e.g. see Figure 1 for a typical experimental setting.

A. BumbleBee experiments

The BumbleBee radar has a 60 degree conical coverage pattern, which allows for placement on ceilings, walls, posts, tree trunks, etc. It transmits 2 million pulses per second, whose reflections off of objects in the detection range are convolved with a reference pulse to capture the movement in the scene. The resulting signal is integrated over a 1000 pulse period, low pass filtered to 100 Hz, and then sampled by a mote at several hundred Hz. The radar is thus capable of providing information for even minute motions with radial velocities between 2.6cm/s and 2.6m/s.

The datasets consist of 33 recording sessions with data sampled over a maximum duration of 130 minutes (longest streak with the same number of people in a room) that we subsample to various shorter framelengths from 20 up to 160 sec. This gives rise, after filtering, to $N = 1482$ samples for the 30 sec framelength clips that we use as our main experimental dataset. We examine and discuss the effect of different framelengths in the results section. The radar has a detection range of 10m omnidirectional and is typically placed in the back corners of the classrooms covering the extend of the room. It is worth noting that interference from neighborhing rooms and adjacent activities within the detection range is present, unreported, and can be seen to create outliers in the data.

B. Methodologies

We compare the proposed approach against two DFT-based feature construction techniques following a modular learning strategy [24] and past work on kernels derived from STFT features for doppler-based activity recognition [25] and complex Gaussian kernels [17]. All approaches employ GPR as the supervised learning technique and we describe and report only the best DFT competing approaches across a range of benchmarks and parameter settings.

DFT with the real Gaussian kernel

The first approach computes moments based on a standard DFT construction and embeds them in a Gaussian kernel (with ARD), similarly to [25], within GPR . Specifically, in this case we compute the STFT of the complex signal using a Chebyshev window and 50% window overlap. In order to set the window length we perform a grid search that includes window sizes that are powers of 2 between 2 and 256. The result is converted to a power spectrum and we compute up to 4th order moments on a per frequency basis. We reduce dimensionality by truncating the DFT at a particular frequency, selecting both the number of frequencies retained, and the maximum order of the moments, by grid search. The resulting moments are embedded in the Gaussian kernel; the vectors have d*m dimensions where d is the number of retained frequencies, and m is the number of moments. It is worth noting that we did not observe an improvement in performance using SVD as an alternate method of dimension reduction.

DFT with the complex Gaussian kernel

In the complex case, we again compute the STFT of the complex signal with the same window function and overlap without converting to power spectrum in order to retain the complex components. Similarly, we compute up to 4th order moments of each frequency; in the complex case this corresponds to a maximum of 8 non-redundant moments, of which 6 are complex-valued. The resulting complex vector is embedded with the complex Gaussian kernel from Equation 2 following [17]. We normalize the diagonal of the kernel to correct for the complex self-similarity measure and proceed with ARD inference for the GPR.

`cString` *kernels*

The string kernel is described in detail in the preceeding sections and summarized in Algorithm 1. In terms of the main parameters we perform 10-fold cross-validation grid search for the alphabet size $|\mathbf{A}|$ from 2×2 up to 14×14, n-gram lengths up to 7 symbol-pairs and PAA window sizes from 1 to 20.

V. Results

We report 10-fold CV results on the 30 second frame-lengths in terms of the correlation coefficient r, RMSE and the coefficient of determination R^2. First we report results from the DFT approaches with both the real and complex kernels and proceed to the proposed symbolic approach. The following notation applies: m is the order of moments employed (e.g. m=2 implies 1st and 2nd moment), n is the n-gram length and window sizes are the PAA window for the symbolic and the Chebyshev window for DFT.

A. DFT with the real Gaussian kernel

In Figure 5(a) the correlation coefficient r is shown for various parameter settings on 10-fold CV with the Gaussian kernel operating on the DFT moments. The corresponding RMSE and R^2 results are shown in Figures 5(b) and 6 respectively. The best performances are $r = 0.89$, RMSE$= 5.16$, $R^2 = 0.8$.

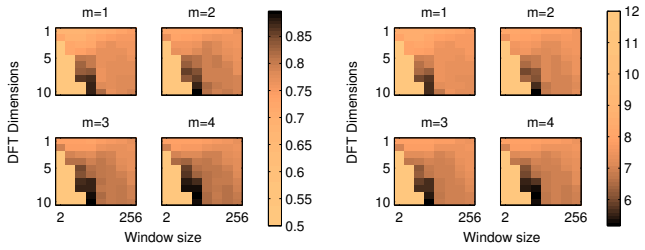

Fig. 5. DFT with power spectrum: r and RMSE

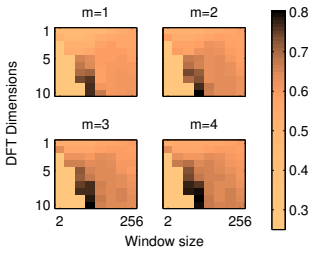

Fig. 6. DFT with power spectrum: R^2

B. DFT with the complex Gaussian kernel

In Figure 7(a) the correlation coefficient r is shown for various parameter settings on 10-fold CV with the Gaussian complex kernel operating on the DFT features. The corresponding RMSE and R^2 results are shown in Figures 7(b) and 8 respectively. The best performances from the complex kernel are $r = 0.84$, RMSE$= 6.26$, $R^2 = 0.71$

C. `cStrings`: String kernel

In Figure 2(a) we plot the estimated versus the real count with the best performing settings for `cSrings` found from 10-fold cross-validation. We find that an alphabet size of 9 for each dimension, a PAA window size of 10 and an 6-gram order give the best performance that outperforms the complex Gaussian kernel and the DFT representation. The best performances from the proposed approach are $r = 0.98$, RMSE$= 2.11$, $R^2 = 0.97$

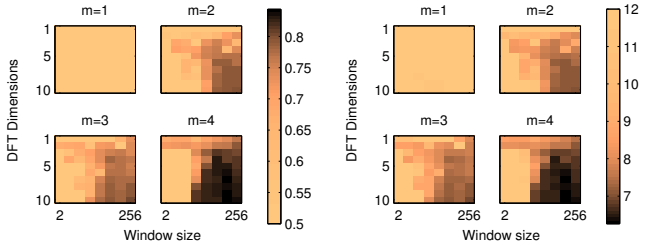

Fig. 7. DFT with complex kernel: r and RMSE

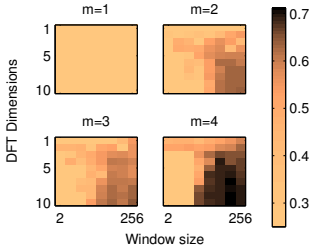

Fig. 8. DFT with complex kernel: R^2

D. Performance stability across parameter settings

In analogy with the competing approaches we examine the stability of the proposed algorithm acrosss different parameter settings. In Figure 9(a) we plot the 10-fold CV estimate of the correlation coefficient across the 3 main parameters for n-gram, alphabet and PAA window sizes. As it can be seen the performance of the estimator is quite stable across settings for $n \geq 4$. Similarly in Figure 10(a) we can see the R^2 values across settings and finally in Figure 9(b) the corresponding RMSE estimates.

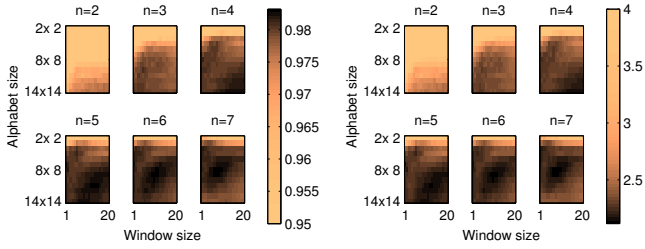

Fig. 9. `cStrings` correlation coefficient and RMSE values

E. Effect of frame length

Finally, we examine the effect of the framelength on the estimator using the best performing parameter settings. As we can see from Figure 10(b) as we increase the length of the time frame performance experiences a small drop which can be explained as the compounding effect of two processes. First, as we increase the framelength different moving "activities" may take place hence adding signal variability. Secondly, the available sample size for 10-CV decreases placing us earlier in the learning curve. Despite that, even the worst performing framelength (160s with $r = 0.96$, RMSE$= 3.06$,

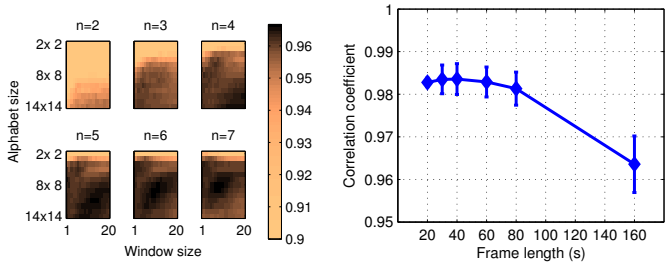

Fig. 10. (a) `cStrings`: Corresponding R^2 values and (b) Framelength effect on best performing `cStrings` estimator

$R^2 = 0.93$) still outperforms the best DFT results. Finally, in terms of tuning the framelength, which is activity and problem dependent, standard cross-validation procedures can be used to learn an optimal setting when the application goal and constraints do not already pre-define the experimental setting and specific predictive task.

VI. Discusion

We proposed a symbolic representation, that we call `cStrings`, for complex time series that is shown to provide state of the art performance ($r = 0.985$) on a challenging task of counting people from a pulsed Doppler radar sensor that detects movement. Our approach significantly outperforms DFT representations and recently proposed complex kernels while offering uncertainty estimates, Figure 2(a), and avoiding the complexity of manipulating DFT spectrograms. The approach extends SAX to the complex domain while retaining a lower bound on distance measures and hence is applicable to general complex time series modelling and alignments. The sensor is subject to masking and superposition effects but we are able to accurately predict the number of targets within a 30s sampling event. We are now examining complexity improvements based on numerocity reduction [8] and improved string kernel implementations while increasing the complexity of the sensed environments and evaluating activity-level recognition. To that front we are collecting data with better coverage of possible activities and movement patterns to reduce the induced frame-level correlation in the existing experimental setting. We believe this work is a significant step towards efficient and accurate target counting from non-vision based radar sensors.

References

[1] J. C. S. Jacques Junior, S. R. Musse, and C. R. Jung, "Crowd analysis using computer vision techniques," *Signal Processing Magazine, IEEE*, vol. 27, no. 5, pp. 66–77, 2010.

[2] A. B. Chan, Z.-S. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE, 2008, pp. 1–7.

[3] A. Treptow, G. Cielniak, and T. Duckett, "Real-time people tracking for mobile robots using thermal vision," *Robotics and Autonomous Systems*, vol. 54, no. 9, pp. 729–739, 2006.

[4] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[5] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, *Fast subsequence matching in time-series databases*. ACM, 1994, vol. 23, no. 2.

[6] D. Q. Goldin and P. C. Kanellakis, "On similarity queries for time-series data: constraint specification and implementation," in *Principles and Practice of Constraint Programming—CP'95*. Springer, 1995, pp. 137–153.

[7] K.-P. Chan and A. W.-C. Fu, "Efficient time series matching by wavelets," in *Data Engineering, 1999. Proceedings., 15th International Conference on*. IEEE, 1999, pp. 126–133.

[8] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," in *Proceedings of the 23rd international conference on Machine learning (ICML-06)*. ACM, 2006, pp. 1033–1040.

[9] M. Cuturi, J.-P. Vert, O. Birkenes, and T. Matsui, "A kernel for time series based on global alignments," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 2. IEEE, 2007, pp. II–413.

[10] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden markov models," *Signal Processing, IEEE Transactions on*, vol. 46, no. 4, pp. 886–902, 1998.

[11] L. Li and B. A. Prakash, "Time series clustering: Complex is simpler!" in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 185–192.

[12] P. Patel, E. Keogh, J. Lin, and S. Lonardi, "Mining motifs in massive time series databases," in *IEEE International Conference in Data Mining (ICDM)*. IEEE, 2002, pp. 370–377.

[13] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD*. ACM, 2003, pp. 2–11.

[14] E. Keogh, J. Lin, and A. Fu, "Hot sax: Efficiently finding the most unusual time series subsequence," in *Data mining, fifth IEEE international conference on*. IEEE, 2005, pp. 8–pp.

[15] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Locally adaptive dimensionality reduction for indexing large time series databases," in *ACM SIGMOD Record*, vol. 30, no. 2. ACM, 2001, pp. 151–162.

[16] B.-K. Yi and C. Faloutsos, "Fast time sequence indexing for arbitrary lp norms." VLDB, 2000.

[17] P. Bouboulis and S. Theodoridis, "Extension of Wirtinger's Calculus to Reproducing Kernel Hilbert Spaces and the Complex Kernel LMS," *IEEE Transactions on Signal Processing 2011*, 2011.

[18] P. Bouboulis, S. Theodoridis, and C. Mavroforakis, "Complex support vector machines for regression and quaternary classification," *CoRR*, vol. abs/1303.2184, 2013.

[19] E. J. Keogh and M. J. Pazzani, "A simple dimensionality reduction technique for fast similarity search in large time series databases," in *Knowledge Discovery and Data Mining. Current Issues and New Applications*. Springer, 2000, pp. 122–133.

[20] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *The Journal of Machine Learning Research*, vol. 2, pp. 419–444, 2002.

[21] C. Leslie and R. Kuang, "Fast string kernels using inexact matching for protein sequences," *The Journal of Machine Learning Research*, vol. 5, pp. 1435–1455, 2004.

[22] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, 2006.

[23] R. Herbrich, N. D. Lawrence, and M. Seeger, "Fast sparse gaussian process methods: The informative vector machine," in *Advances in neural information processing systems*, 2002, pp. 609–616.

[24] S. Haykin and T. K. Bhattacharya, "Modular learning strategy for signal detection in a nonstationary environment," *Signal Processing, IEEE Transactions on*, vol. 45, no. 6, pp. 1619–1637, 1997.

[25] Y. Kim and H. Ling, "Human activity classification based on micro-doppler signatures using a support vector machine," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 47, no. 5, pp. 1328–1337, 2009.